1990-12

A ring model for local mobile radio communications with variable packet length

Banh, Dennis V.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/27549

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A RING MODEL FOR LOCAL/MOBILE RADIO COMMUNICATIONS WITH VARIABLE PACKET LENGTH

by

Dennis V. Banh

December 1990

Thesis Advisor: Tri T. Ha
Co-Advisor: R. Clark Robertson

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): EC	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) A RING MODEL FOR LOCAL/MOBILE RADIO COMMUNICATIONS WITH VARIABLE PACKET LENGTH

12 PERSONAL AUTHOR(S) BANH, Dennis V.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1990 December	57

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	a ring model for local/mobile radio communications; variable packet length; near/far effect; equal ring throughput

19 ABSTRACT (Continue on reverse if necessary and identify by block number) This thesis presents an analysis of the performance of a local/mobile radio communications system utilizing the Aloha random access protocol with variable length packets. The capture phenomenon due to the near/far effect that enhances the performance of the system is investigated. A tagged packet will capture the base station if its signal-to-interference ratio exceeds a threshold γ_o. Because of the near/far effect, users near the base station typically have a stronger signal than those farther away. A multiple ring model is used to alleviate this problem. Users in one ring employ different retransmission strategy from those in other rings. A shorter retransmission delay is allocated to users in rings farther from the base station than in those closer to the base station in order to achieve approximately the same average delay throughout the network regardless of location.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL: HA, Tri T./Robertson, R. Clark	22b TELEPHONE (Include Area Code): 408-646-2788/2383; 22c OFFICE SYMBOL: EC/Ha EC/Rc

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603 UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

A Ring Model for Local/Mobile Radio Communications with Variable Packet Length

by

Dennis V. Banh
B.S., Cal. State Poly University, Pomona, 1987

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1990

Author: [redacted]
Dennis V. Banh

Approved by: [redacted]
Tri T. Ha, Thesis Advisor

[redacted]
R. Clark Robertson, Co-Advisor

[redacted]
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis presents an analysis of the performance of a local/mobile radio communications system utilizing the Aloha random access protocol with variable length packets. The capture phenomenon due to the near/far effect that enhances the performance of the system is investigated. A tagged packet will capture the base station if its signal-to-interference ratio exceeds a threshold γ_0. Because of the near/far effect, users near the base station typically have a stronger signal than those farther away. A multiple ring model is used to alleviate this problem. Users in one ring employ different retransmission strategy from those in other rings. A shorter retransmission delay is allocated to users in rings farther from the base station than in those closer to the base station in order to achieve approximately the same average delay throughout the network regardless of location.

TABLE OF CONTENTS

I. INTRODUCTION

Aloha protocol is a random access protocol in which users transmit their packets at will. Because of the random times at which users may decide to transmit, there is a chance that two or more users transmit at overlapping times. This results in a collision. Whether the overlapping covers one bit or a whole packet, the collided or overlapped packets will be discarded. Aloha protocol has inherent characteristics of instability. As the traffic intensity increases, the probability of collision increases as well, leading to possible instabilities of the system. Throughput is limited to some maximum value less than the channel capacity. The particular values depend on the access mechanism. For example, the maximum value for pure Aloha is $0.5e^{-1}$ or 0.18. Despite its poor throughput and its inherent instability, the Aloha protocol is still a prime candidate for many local and mobile packet radio systems because of its ease in implementation. In many practical applications, Aloha performance in local and mobile communication systems is affected by the near/far effect as well as multi-path fading. The near/far effect arises when users are located at different distances from the base station. The power levels of the received packets may vary substantially due to spatial attenuation of the signals, and the packet with highest power level has the best chance of

being received correctly. Multi-path fading effects also cause random power variation among the received packets. Hence, the near/far effect and multi-path fading combine to allow a receiver to correctly receive a packet even when multiple packets are transmitted simultaneously. Recent research has shown that each of the above effects do improve the throughput of an Aloha channel considerably [1-3]. However, this is achieved in favor of the users near the base station and at the expense of the far users. The far users have to retransmit their packets more often than the near users and hence suffer a longer packet delay.

To correct this problem, a ring model in which the coverage area is divided into 3 concentric rings with the base station in the center is investigated in this thesis. Users in different rings use different strategies to retransmit their collided packets. A ring that is far from the base station will have a shorter average retransmission delay than a ring that is closer to the base station to compensate for the higher capture probability of the near users. As a result, users in a ring model may achieve approximately the same packet delay.

This thesis presents an analysis of the near/far effect on channels employing variable packet Aloha in a local/mobile radio ring model. Unlike idealized Aloha in which a tagged packet is only received successfully when it does not collide with other packets, the capture effect enables the base

station to accurately decode the tagged packet even when it is involved in a collision. The result of the analysis is used to find the equal ring-throughput of a network consisting of three concentric rings.

Chapter II presents a typical system model of local/mobil radio communications. Chapter III analyzes the dynamics of number of interfering packets in variable packet Aloha channels. The probability of capture of a tagged packet by the base station and the system throughput will be established in chapter III. Finally, in chapter IV a network of three rings is analyzed to find the ring widths such that the throughput in each ring is approximate equal.

II. SYSTEM MODEL

The system model is a network consisting of users assumed to be uniformly distributed in an annular ring surrounding the base station. Figure 1 depicts the system model. Assume that the user traffic density is $g(r)$, where r is the distance to the base station. The density g(r) can be thought of as the normalized packet traffic per unit area at a distance r from the base station. The packet generation rate in a given area depends on r and is independent of direction. Then the traffic in a differential ring of width dr at a radius r from the base station is [4]

$$G(dr) = 2\pi r g(r)\, dr \tag{1}$$

The traffic generated within a distance r of the base station is

$$G(r) = 2\pi \int_0^r x g(x)\, dx \tag{2}$$

The total offered traffic in the network can be obtained as G = G(r → ∞)

$$G = 2\pi \int_0^\infty x g(x)\, dx \tag{3}$$

The spatial distribution of packet generation is found to be the probability a packet is generated within a distance $R \le r$

$$F_R(r) = Pr\{R \le r\} = \frac{2\pi}{G} \int_0^r x g(x)\, dx \tag{4}$$

Therefore, the probability density function for packet generation is found to be

$$f_R(r) = \frac{2\pi}{G} r g(r) \tag{5}$$

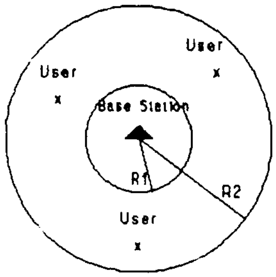

Figure 1. The system model

Assume that users are uniformly distributed in an annular ring $R_1 < r \le R_2$ around the base station, then

$$g(r) = \begin{cases} \dfrac{G}{\pi (R_2^2 - R_1^2)} & , R_1 \le r \le R_2 \\ 0 & , otherwise \end{cases} \tag{6}$$

Substituting (6) into (5), we obtain

$$f_R(r) = \begin{cases} \dfrac{2r}{R_2^2 - R_1^2} & , R_1 \le r \le R_2 \\ 0 & , otherwise \end{cases} \tag{7}$$

Figure 2 shows the distribution of packet generation of a ring with $R_1 \leq r \leq R_2$.

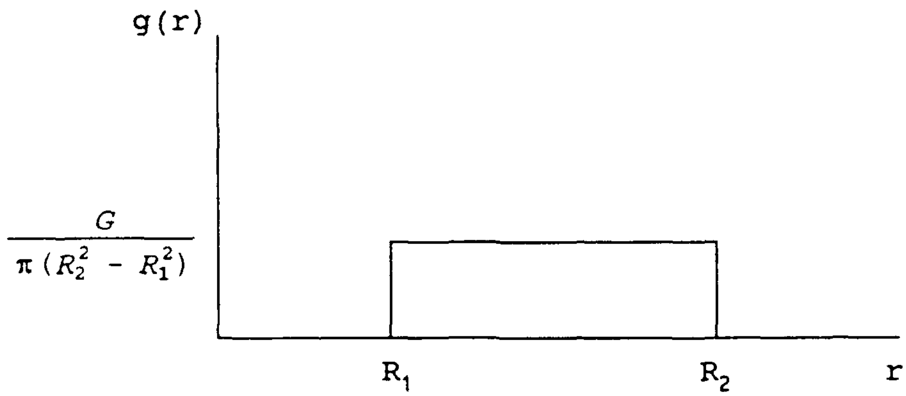

Figure 2. User traffic density g(r)

There are two main effects that influence the performance of local/mobile radio communications, namely the near/far effect and multi-path fading. To model the near/far effect, assume that signal power depends on the spatial distribution of users in the network. The mean power of a packet at a distance r from the base station is of the general form

$$w = cr^{-\alpha} \tag{8}$$

where α is between 3 and 5 for ultra high frequency (UHF) applications. The constant c depends on transmitted power and the gains and the heights above the ground of the base station and mobile antennas. Assuming all users are identical we can normalize the constant c to unity for all users. Hence,

$$w = r^{-\alpha} \tag{9}$$

Let the power random variable W be defined by

$$W = R^{-\alpha} \tag{10}$$

The event $\{W \le w\}$ occurs when event $A = \{R^{-\alpha} \le w\}$ occurs. A can be rewritten as $A = \{R \ge w^{-1/\alpha}\}$. Thus,

$$F_W(w) = P[R \ge w^{-1/\alpha}] \tag{11}$$

Therefore, (11) is equivalent to

$$F_W(w) = 1 - F_R(w^{-1/\alpha}) \tag{12}$$

Differentiate both side of (12)

$$f_W(w) = -f_R(w^{-1/\alpha}) \frac{d(w^{-1/\alpha})}{dr} \tag{13}$$

Expanding (13), we get

$$f_W(w) = f_R(w^{\frac{-1}{\alpha}}) \frac{w^{\frac{-1}{\alpha}-1}}{\alpha} \tag{14}$$

Substituting (7) into (14), we obtain

$$f_W(w) = \frac{2}{\alpha(R_2^2 - R_1^2)} w^{-(\frac{2}{\alpha}+1)} \tag{15}$$

For a typical value $\alpha = 4$ in UHF communications, (15) becomes

$$f_W(w) = \frac{w^{-(3/2)}}{2(R_2^2 - R_1^2)} \tag{16}$$

One of the important factors in determining the quality of signal reception is the number of interferers to the tagged packet. Thus, it is of interest to analyze the dynamics of the system. This is presented in the next chapter.

II. SYSTEM DYNAMICS

In this study, we propose variable packet Aloha instead of the conventional fixed length packet Aloha. Variable packets do not require padded bits to fill up a packet to its specified length which may reduce the throughput of fixed packet Aloha. Variable packets also do not require bit stuffing and de-stuffing logic which may increase the cost of hardware. The channel traffic rate is assumed to obey a Poisson process with parameter g packets per second. The newly created packets are of variable length x (seconds) and are independently and identically distributed according to the probability density function a(x) with mean $\bar{x}$. All retransmitted packets have their length x redrawn afresh from a(x) for every retransmission.

The probability that n packets interfere (overlap) with an arbitrary tagged packet during its transmission is given by [5]

$$Pr\{n\} = \frac{[g(x+u)]^n}{n!} e^{-g(x+u)} \tag{17}$$

where x is the tagged packet length, u is the random length of a preceding packet, and n is the number of interfering packets.

The probability that the tagged packet encounters no overlap from previous packets and the next inter-arrival is

larger than x is given by n = 0. This is also the probability of successful tagged packet transmission for the case where no interfering packets can be tolerated by a tagged packet.

This analysis considers the case when the base station is captured by a tagged packet in the presence of interfering packets if the ratio of the tagged packet signal power and the joint interfering packets signal power exceeds a threshold γ_0.

The number of interfering packets in a variable packet Aloha channel is a stochastic process which is driven by four types of interferers. The first type of interferers are designated as Early-Late packets. These are packets that begin before the tagged packet and end after the tagged packet. The remaining types of interferers have similar descriptive nomenclatures and are designated as Early-Early packets, Late-Early packets, and finally Late-Late packets. Figure 3 depicts four types of interferers relative to the tagged packet.

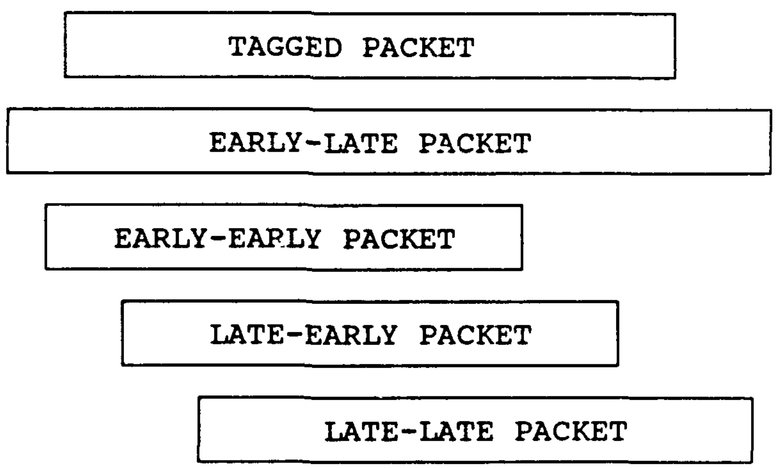

Figure 3. Four types of interfering packets

The arrival and departure of n interfering packets partition the tagged packet into several non-overlapping intervals of random length. The number of these intervals, given a known number of interfering packets n and their corresponding type of interfering packets found in is [6]

$$\textit{Total Intervals} = \textit{number of Early-Early packets} + 2 \times (\textit{number of Late-Early packets}) + \textit{number of Late-Late packets} + 1 \quad (18)$$

The relationship in (18) is based on the following facts. There is initially one interval, the tagged packet itself, when there are no interfering packets. The presence of a single Early-Early packet divides the tagged packet into two smaller intervals; it adds one more to the total number of intervals. The presence of a single Late-Late packet also partitions the tagged packet and provides one more to the total number of intervals. The presence of a single Late-Early packet will further divide the tagged packet and adds two more to the total number of intervals. The Early-Late packet overlaps entire duration of the tagged packet and therefore contributes no extra intervals to the total number of intervals.

Each interval has associated with it a total number of interferers $I(t_i)$, where t_i is the start time of the i'th interval of the tagged packet. Each unique sequence of total interferers per partition $I(t_i)$ for all i is called a realization. Figure 4 depicts a specific realization.

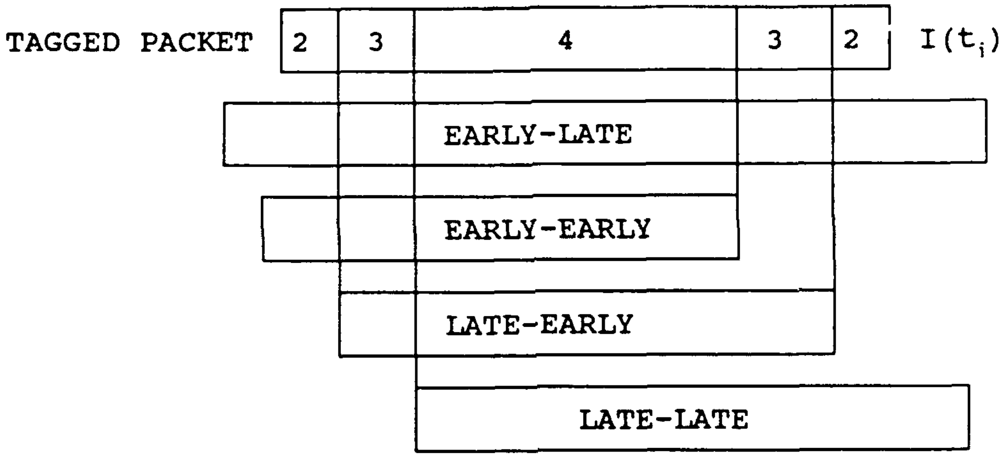

Figure 4. An example of a specific realization. The total number of interferers $I(t_i) = \{2,3,4,3,2\}$

For this specific realization, the number of interfering packets is four. There is one of each type of packet: one Early-Late packet, one Early-Early packet, one Late-Early packet, and one Late-Late packet. The total number of intervals during the tagged packet according to (18) is five. The total number of interferers for each interval is shown at the top of Figure 4. We are interested in finding every possible realization for a given n users. From those realizations, the number of realizations C_j in which the maximum number of interferers present at any time during the tagged packet interval does not exceed some number j, will be calculated.

Finding all possible realizations is very tedious but is easily obtained via computer. TABLE I lists the algorithm of the main program.

TABLE I. MAIN PROGRAM ALGORITHM

```
% N+1 is a number of users in the network
for n = 1:N,
% call subroutine make_comb to find all possible
% combination of interfering packets.
    length = make_comb();
    for i = 1:length,
% call subroutine realization to build a binary
% tree of all possible realizations
        realization();
% call get_cj to count number of realizations
% which has maximum number of interferer not
% exceeding j
        get_cj();
    end
end
```

For a given n from 1 to N, where N is the maximum number of interferers in the system, subroutine make_comb finds all possible combinations of four types of packets taken n at a time. Basically, subroutine make_comb assigns different values each time for each type of interfering packets such that the total of all interfering packets is equal to n and enumerates the number of distinctive combinations found. Subroutine make_comb uses the algorithm found in [7] to generate all the combinations. For example, given n = 2, the set of combinations of four types of interfering packets will be U = {(2,0,0,0), (0,2,0,0), (0,0,2,0), (0,0,0,2), (1,1,0,0), (1,0,1,0), (1,0,0,1), (0,1,1,0), (0,1,0,1), (0,0,1,1)}.

For each combination of four packet types taken n at a time, the subroutine realization is called to find the total number of realizations. The root of the binary tree corresponds to the first interval and contains the initial

number of interferers. In the first interval, the interferers can only be those packet types beginning before the tagged packet; i.e., Early-Late packet(s) and Early-Early packet(s). The initial number of interferers will be the sum of those packets. The children of the root of the binary tree corresponds to interval 2 and their children corresponds to interval 3 and so on for the rest of all intervals. The next intervals of the tagged packet contain either one packet more or one packet less than the number of interfering packets of the previous interval because their existence depends on either the arrival or the departure of one of the interfering packets. However, the last interval should be equal to number of those packets ending after the tagged packet ends. We choose to implement the binary tree by having the left child node taken care of the case of arrival of an interfering packet and the right child node for the case of departure of an interfering packet. In other words, the number of interferers stored at a left child node of a parent node is equal to the parent's number of interferers plus one if the parent node is not greater than n, the maximum number of interferers. If it is greater than n , that left child node is set to NULL; i.e., any realization containing that node is invalid. The number of interferers stored at a right child is equal to the parent node's minus one if the number of interferers of the parent node is greater than the minimum number of interferers which is equal to number of Early-Late

packet(s). If it is not greater than the minimum number of interferers then the right child node is set to NULL. Any realization which has the number of interferers in its last interval not equal to the sum of the number of Early-Late packets and Late-Late packets is also invalid. Figure 5 shows a binary tree of a specific combination of four types of interfering packets (1,1,1,1).

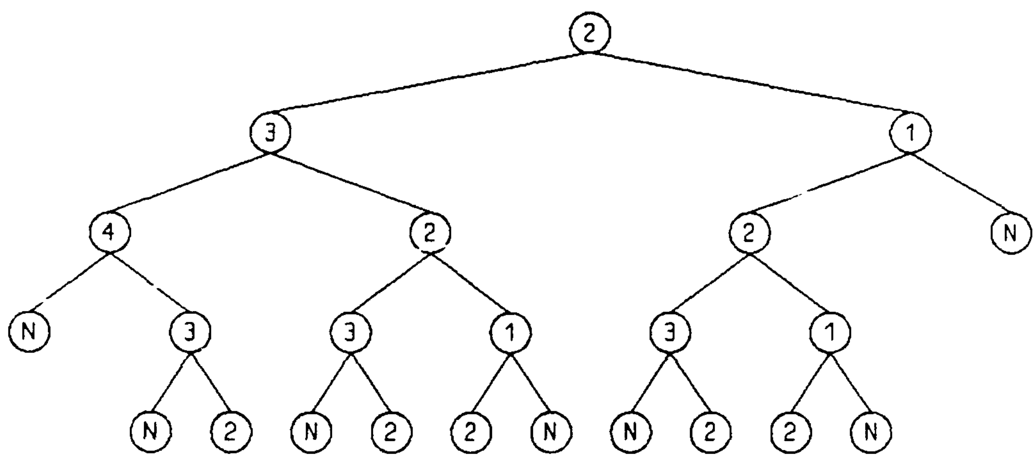

Figure 5. A binary tree implementation of (1,1,1,1)

The subroutine get_cj traverses the binary tree to collect all the valid realizations. The count of realizations which have the same number of maximum number of interferers j at any time is C_j.

TABLE II shows the output of the computer program for n =1 to 7. For the case of n = 1, there are only four possible realizations, namely the interferer can be Early-Early, Early-Late, Late-Early, or Late-Late packet. Each realization has

only one known interferers. Therefore, the value of $C_1(1)$ is four. Other values of $C_j(1)$ such as $C_2(1)$, $C_3(1)$, etc. are 0. For the case of n = 2, there are 10 combinations of four packet types taken two at a time. These combinations generate 14 unique realizations. Of these 14 realizations, four have a maximum number of interferers of 1 at any time, and 10 have a maximum number of interferers of two at any time. The value of $C_1(2)$ is 4 and $C_2(2)$ is 10. Other $C_j(2)$ values are 0. The function I_T is the total number of all possible realizations for n known interfering packets. The function f(n) is the inverse of I_T.

TABLE II. MAXIMUM NUMBER OF INTERFERING PACKETS GIVEN n

n	C_1	C_2	C_3	C_4	C_5	C_6	C_7	I_T	f(n)
1	4	0	0	0	0	0	0	4	0.25
2	4	10	0	0	0	0	0	14	0.0714
3	4	24	20	0	0	0	0	48	0.0208
4	4	50	75	35	0	0	0	164	0.0061
5	4	100	225	176	56	0	0	561	0.0018
6	4	200	625	664	350	84	0	1927	0.0005
7	4	400	1698	2250	1565	624	120	6661	0.0002

III. PROBABILITY OF CAPTURE AND THROUGHPUT OF VARIABLE PACKET ALOHA CHANNELS

A. PROBABILITY OF CAPTURE

As we have mentioned earlier, the near/far effect arises when users are scattered at different distances from the base station and causes the power levels of received packets to vary substantially. Thus, the arriving packet with the highest power level has a best chance of being received correctly. This model in which a variable packet Aloha protocol is employed has better throughput than the ideal Aloha protocol due to the near/far effect. A tagged packet has the capability to capture the base station even when it is involved in a collision. We assume that a tagged packet will capture the base station if its signal-to interference ratio exceeds a threshold γ_o. We also assume that thermal noise is negligible since it is small compared to the competing interference of interfering packets. In this section we calculate the probability of capture for the variable packet Aloha ring model and develop an expression for the throughput.

Let the random variable X denote the power of a tagged packet with the probability density function (pdf) $f_X(x)$, and the random variable Y denote the joint interfering power of n ≥ 1 interfering packets with the conditional probability

density function $f_Y(y|n)$. With the assumption of independence, the conditional pdf $f_Y(y|n)$ is found by the relationship [6]

$$f_Y(y|n) = f(n) \sum_{j=1}^{n} C_j(n) \, [f_W(w)]^{j\otimes} \tag{19}$$

where $j\otimes$ denotes j-fold convolution, f(n) is the inverse of total number of realizations given n interfering packets, and $C_j(n)$ is the number of realizations in which a maximum number of interfering packets equals j given n interferers. Both f(n) and $C_j(n)$ are listed in TABLE II.

Let the random variable Z be the signal-to-interference ratio

$$Z = \frac{X}{Y} \tag{20}$$

then the probability density function of Z is [8]

$$f_Z(z|n) = \int_0^{\infty} y f_{X,Y}(x,y|n) \, dy \tag{21}$$

Given the fact that X and Y are independent random variables, we obtain

$$f_Z(z|n) = \int_0^{\infty} y f_X(yz) \, f_Y(y|n) \, dy \tag{22}$$

The probability of capture, that is the probability that Z is greater than a threshold γ_0 given n interfering packets

$$Pr\{capture|n\} = Pr\{\gamma_0 \le Z \le \infty|n\} \tag{23}$$

is obtained by integrating $f_Z(z|n)$ from γ_0 to ∞. Hence

$$Pr\{capture|n\} = \int_{\gamma_0}^{\infty} f_z(z|n)\, dz \tag{24}$$

The overall probability of capture of a tagged packet is found by multiplying the conditional probability of capture given n interfering packets by the probability that n packets interfere with a tagged packet during its transmission, and summing over all n.

$$Pr\{capture\} = \sum_{n=0}^{N} Pr\{capture|n\}\, Pr\{n\} \tag{25}$$

where N+1 is the number of users in the system.

Substituting (17) into (25), we obtain

$$Pr\{capture\} = \sum_{n=0}^{N} \frac{[g(x+u)]^n}{n!} e^{-g(x+u)}\, Pr\{capture|n\} \tag{26}$$

For the purpose of illustration, consider a network with $R_1 = 0$ and $R_2 = 1$. The pdf of packet power f_w in (16) becomes

$$f_w(w) = \frac{1}{2} w^{-(3/2)} \tag{27}$$

Equation (21) and the conditional probability of capture Pr{capture|n} with the assumption of γ_0 being equal to 6 dB is obtained numerically using Pro-Matlab (see Appendix A for the program). TABLE III shows the conditional probability of capture for n from 0 to 7.

In the next section we will incorporate results in TABLE III to find the throughput of a variable packet Aloha channel in a ring model.

TABLE III. THE CONDITIONAL PROBABILITY OF CAPTURE

n	Pr{capture|n}
0	1.0000
1	0.2002
2	0.1503
3	0.1202
4	0.1018
5	0.0894
6	0.0806
7	0.0740

B. THROUGHPUT OF A VARIABLE PACKET ALOHA RING MODEL

The channel throughput S, in packets per mean packet time is [5].

$$S = \int_0^\infty \int_0^\infty \frac{1}{2}\phi(x,u)\, a(x)\, a(u)\, dx\, du \tag{28}$$

where a(x) and a(u) are exponential density functions with mean $\overline{x}$ of the tagged packet length and the preceding packet length, respectively. The quantity $\phi(x,u)/2$ is the conditional channel departure rate [5].

$$\frac{1}{2}\phi(x,u) = \frac{1}{2} g(x+u)\, Pr(capture) \tag{29}$$

Substituting (26) and (29) into (28), we have

$$S = \int_0^\infty \int_0^\infty \frac{1}{2} g(x+u) \sum_{n=0}^{\infty} e^{-g(x+u)} \frac{[g(x+u)]^n}{n!} \cdot Pr\{capture|n\}\, a(x)\, a(u)\, dxdu \qquad (30)$$

Utilizing the binomial identity

$$(x+u)^n = \sum_{k=0}^{n} \binom{n}{k} x^k u^{n-k} \qquad (31)$$

and expanding (30), we obtain

$$S = \frac{1}{2}\int_0^\infty \int_0^\infty [gxe^{-gx} e^{-gu} + gue^{-gu} e^{-gx}] \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\} \cdot \sum_{k=0}^{n} \binom{n}{k} x^k u^{n-k}\, a(x)\, a(u)\, dxdu \qquad (32)$$

Further manipulating terms in the above equation gives

$$S = \frac{1}{2} g \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\} \cdot \left[\sum_{k=0}^{n} \binom{n}{k} \int_0^\infty x^{k+1} e^{-gx} a(x)\, dx. \int_0^\infty u^{n-k} e^{-gu} a(u)\, du + \sum_{k=0}^{n} \binom{n}{k} \int_0^\infty u^{n-k+1} e^{-gu} a(u)\, du. \int_0^\infty x^k e^{-gx} a(x)\, dx\right] \qquad (33)$$

We assume that the packet length is exponentially distributed and the average length is $\bar{x}$. Then we have

$$a(x) = \frac{1}{\bar{x}} e^{-x/\bar{x}} \qquad (34)$$

$$a(u) = \frac{1}{\bar{x}} e^{-u/\bar{x}} \qquad (35)$$

Define the average traffic rate G as

$$G = g\bar{x} \qquad (36)$$

We apply (34), (35), (36) and [9].

$$\int_0^{\infty} x^n e^{-\mu x}\, dx = n!\,\mu^{-(n+1)} \tag{37}$$

to solve (33)

The first integral term is evaluated as

$$\begin{aligned}\int_0^{\infty} x^{(k+1)} e^{-gx} a(x)\, dx &= \frac{1}{\overline{x}}\int_0^{\infty} x^{(k+1)} e^{-(g+\frac{1}{\overline{x}})x} dx \\ &= \frac{(k+1)!}{\overline{x}} (g+\frac{1}{\overline{x}})^{-(k+2)} \\ &= \frac{(k+1)!}{\overline{x}^{-(k+1)}} (G+1)^{-(k+2)}\end{aligned} \tag{38}$$

Similarly, the second integral is found to be

$$\begin{aligned}\int_0^{\infty} u^{(n-k)} e^{-gu} a(u)\, du &= \frac{1}{\overline{u}}\int_0^{\infty} u^{(n-k)} e^{-(g+\frac{1}{\overline{u}})u} du \\ &= \frac{(n-k)!}{\overline{x}} (g+\frac{1}{\overline{x}})^{-(n-k+1)} \\ &= \frac{(n-k)!}{\overline{x}^{-(n-k)}} (G+1)^{-(n-k+1)}\end{aligned} \tag{39}$$

The third integral term is

$$\int_0^{\infty} u^{(n-k+1)} e^{-gu}\, a(u)\, du = \frac{(n-k+1)!}{\overline{x}^{-(n-k+1)}} (G+1)^{-(n-k+2)} \tag{40}$$

The final integral term is reduced to

$$\int_0^{\infty} x^n e^{-gx} a(x)\, dx = \frac{k!}{\overline{x}^{-k}} (G+1)^{-(k+1)} \tag{41}$$

Combining all integral terms in (38), (39), (40), and (41) into (33), we get

$$S = \frac{1}{2} g \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\} \frac{(G+1)^{-(n+3)}}{\overline{x}^{-(n+1)}} \sum_{k=0}^{n} n!\,(n+2) \tag{42}$$

Applying (36) and the equality

$$\sum_{k=0}^{n} n!\,(n+2) = n!\,(n+1)\,(n+2) \tag{43}$$

and further simplifying (42), we obtain

$$S = \sum_{n=0}^{\infty} \frac{(n+1)\,(n+2)}{2} Pr\{capture|n\}\, G^{n+1}(G+1)^{-(n+3)} \tag{44}$$

For the network with $R_1 = 0$, $R_2 = 1$, and the conditional probability of capture in Table III, throughput is shown in Figure 6.

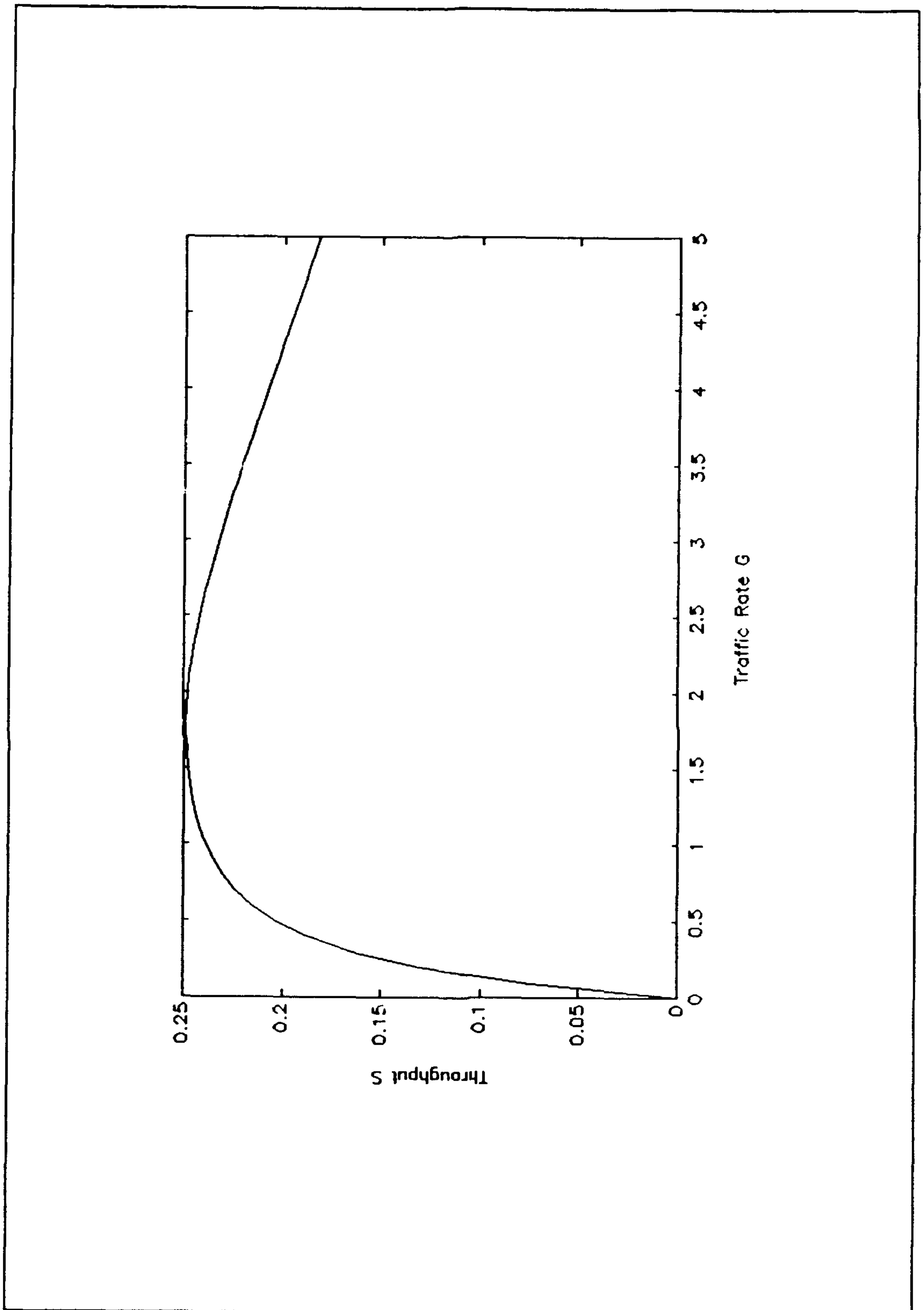

Figure 6. Throughput of a ring model

IV. APPROXIMATE EQUAL RING THROUGHPUT

In this section, we expand our analysis to a system of multiple rings. As we mentioned before, the probability of capture favors near users at the expense of far users resulting in lower throughput for the far users. A multiple ring system compensates for this deficiency by using separate retransmission strategies for users in different rings. A user in a ring that is far from the base station is given a shorter average transmission delay than a user in a ring closer to the base station to compensate for a higher capture probability of the near ring.

A. PROCEDURE TO FIND EQUAL RING THROUGHPUT

A procedure analogous to the procedure found in [3] is used to find the ring width for each ring such that the throughput in each ring is approximately equal. The procedure consists of the following iterative process:

1. Select initial ring widths $R_i = \{r_i < r \le r_{i+1},\}$, where $i = 1,2,\ldots,M$.

2. Find the conditional ring capture probability $\Pr\{capture|n,R_i\}$ (see Appendix B).

3. Find the traffic rate G_i in each ring

$$G_i = G[F_R(r_{i+1}) - F_R(r_i)] \qquad (45)$$

where $i = 1,2,\ldots,M$ and $F_R(r_i)$ is given in (B3)

4. Find the ring throughput for each ring from (Appendix C).

$$S_i = \sum_{n=0}^{\infty}\left[\frac{(n+1)(n+2)}{2}Pr\{capture|n,R_i\}G_i G^n (G+1)^{-(n+3)}\right] \quad (46)$$

5. If $S_1 \approx S_2 \approx \ldots \approx S_M$ for $0 \le S_i \le \min\{\max S_i\}$ go to next step. Otherwise, go back to step 1. For our computational purpose, the maximum relative difference of ±0.05 is acceptable for the approximate equality.

6. Calculate the total throughput of the model $S = \Sigma S_i$.

For illustrative purpose, consider a system of three rings $R_1 = \{0 < r \le 0.52\}$, $R_2 = \{0.52 < r < 0.8\}$ and $R_3 = \{0.8 < r \le 1\}$. The threshold γ_0 is taken to be 6 dB. Fig 7 shows the ring throughput of ring R_1, R_2, R_3 and the total ring throughput of the model. The ring throughputs are approximately equal for $0 < G \le 0.25$.

B. AVERAGE RETRANSMISSION DELAY

To determine the average retransmission delay T_i for ring R_i, we need first to define a retransmission strategy when a collision occurs. One suggestion [10] is to choose an arbitrary time interval and select a uniformly distributed random retransmission time within that interval. Let the time interval covers K_i message-unit times where each message-unit is an average $\overline{x}$ units of time. Then retransmission takes place within 1 to K_i time units after a collision has occurred. The

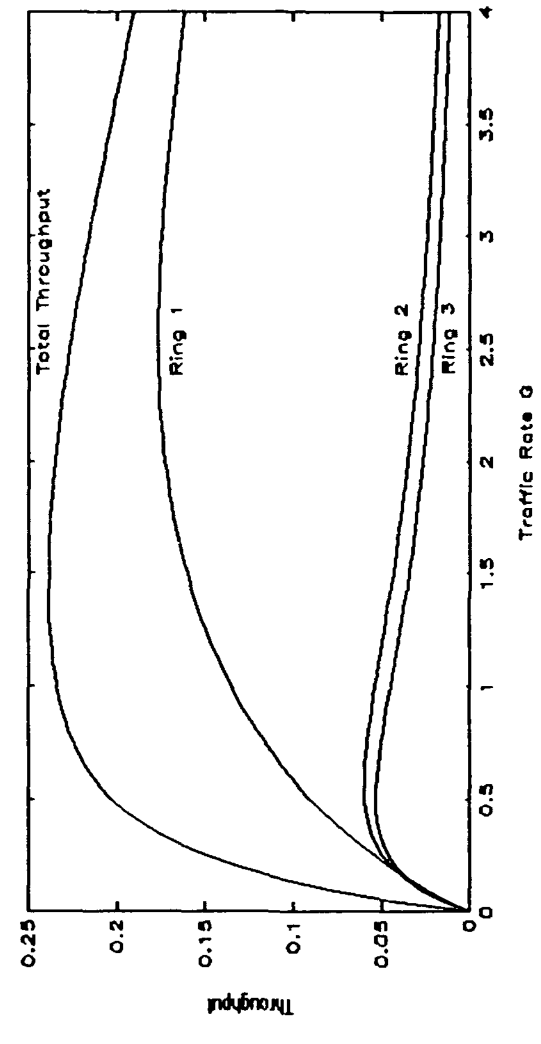

Figure 7. Approximate Equal Ring Throughput

approximate number of retransmissions is found to be

$$E \approx \frac{1-Pr\{capture|R_i\}}{Pr\{capture|R_i\}} \tag{47}$$

where $Pr\{capture|R_i\}$ is the probability of capture in ring R_i per mean packet time is found by (Appendix D)

$$Pr\{capture|R_i\} = \sum_{n=0}^{\infty} Pr\{capture|n,R_i\}\,(n+1)\,G^n\,(G+1)^{-(n+2)} \tag{48}$$

The average retransmission time, ignoring propagation delay, is approximately

$$T_i \approx 0.5E(K_i+1) \tag{49}$$

By fixing the value K_M of the outermost ring, the remaining value K_i can be obtained by letting $T_i = T_M$ and substituting (47) into T_i and T_M expressions, we obtain

$$K_i = \frac{Pr\{capture|R_i\}\,(1-Pr\{capture|R_M\})\,(K_M+1)}{Pr\{capture|R_M\}\,(1-Pr\{capture|R_i\})} - 1 \tag{50}$$

TABLE IV shows the number of retransmission K_1 and K_2 of ring R_1 and ring R_2, given $K_3 = 8$ of $\overline{x}$ time-units. For $0 \leq G \leq G_{max} = 0.25$, the average $\overline{K}_1$ is 29.37 and $\overline{K}_2$ is 8.70.

TABLE IV. VALUES OF K_i

G	K_1	K_2	K_3
0.05	27.9389	8.6379	8
0.10	28.6474	8.6694	8
0.15	29.3634	8.7003	8
0.20	30.0863	8.7306	8
0.25	30.8158	8.7603	8

V. CONCLUSION

We have evaluated the probability of capture taking into account the near/far effect. These results are then used to compute the throughput of a ring model utilizing the variable Aloha protocol in a local/mobil radio communications. We have investigated a ring model in which different retransmission strategies are used to alleviate the near/far effect problem. By assigning shorter average retransmission delay to users farther from the base station and longer average transmission delay to users nearer to the base station, all users may achieve approximately the same average retransmission delay. In this thesis, we only investigated a system of three rings. Further work is needed to determine the number of rings that produces the maximum throughput. The procedure for finding equal ring throughput is a process of trial and error and requires greater effort when a number of rings is greater than three. The capture measure using signal-to-interference threshold provides a less accurate prediction of channel throughput than does a capture measure using packet correction reception probability [1]. However, employing packet correction reception probability as a capture measure for variable packets is very difficult. It is an unwielding task to find the probability of bit error for variable length packets.

APPENDIX A. COMPUTER PROGRAM LISTINGS

A. PROGRAM TO FIND THE MAXIMUM NUMBER OF INTERFERING PACKETS

```
/* The program is written in Unix C to find the number */
/* of realizations Cj which has the maximum number of */
/* interferers equal to j. It first calls subroutine */
/* make_comb to find all the combinations of four types */
/* of interfering packets taken n at a time. Then, for */
/* each  combination it calls subroutine realization to */
/* build a binary tree of realizations. */
/* Finally, subroutine get_cj is called to find Cj. */

#include <stdio.h>
#define comb_lgth 230
#define type 4
#define cj_lgth 10
#define INVALID -1
#define NODE_SIZE sizeof(struct node)

/* Global variables declaration */

   struct node
     {
       int num_intfr;            /* number of interfering
                              packets */
       struct node *left; /* left pointer of a node in a
                                binary tree */
       struct node *right; /* right pointer of a node in
                             a binary tree*/
     };
   FILE *file;         /* file descriptor */
   int num_user;       /* number of users in system */
   int init_intfr;     /* number of intefering packets in
                        the first interval i.e. number of
                        early beginning packets */
   int final_intfr;    /* number of late ending packets */
   int max_intfr;      /* number of maximum possible
                        interfering packets i.e. number of
                        users */
   int min_intfr;      /* minimum number of interfering
                        packets */
                  /*i.e. number of early-late packets */
   int num_interval; /* number of intervals of tagged
                           packet */
   int cj[cj_lgth][cj_lgth];    /* array of Cj */
```

```
/* Function declaration */
   int make_comb();
   struct node *realization();
   void get_cj();
   void print_out();
   char *malloc();

main()
  {
    int i,j;
    int length;         /* length of array comb */
    int comb[comb_lgth][type];/*array of combination of
                                possible packet types*/
    struct node *root;

    file = fopen("datafile","a");
    root = (struct node *)malloc(NODE_SIZE);
    root->left = NULL;
    root->right = NULL;
    /* clear array cj */
    for(i=0;i<cj_lgth;i++)
      {
        for(j=0;j<cj_lgth;j++)
          cj[i][j] = 0;
      }
    for(num_user=1;num_user<=10;num_user++)
      {
    length = make_comb(comb);
    for(i=0; i<=length; i++)
      {
        init_intfr = comb[i][0]+comb[i][1];
        final_intfr = comb[i][0]+comb[i][type-1];
        max_intfr = num_user;
        min_intfr = comb[i][0];
        num_interval = comb[i][1] +
            2*comb[i][2]+comb[i][type-1]+1;
        root->num_intfr = init_intfr;
        root = realization(root);
        get_cj(root);
      }
      }
    print_out();
    fclose(file);
  }

/* The subroutine make_comb is to find all possible */
/* combinations of four types of interfering packets */
/* taken n at a time. */

int make_comb(comb)
int comb[comb_lgth][type];
```

```
{
  int pkt_type[type];
  int length = 0;
  int index_track = -1;
  int num_track;
  int i, j;

  pkt_type[0] = num_user;
  num_track = num_user;
  comb[0][0] = pkt_type[0];    /* first combination  */

  for (i=1; i<type; i++)
    pkt_type[i] = comb[length][i] = 0;
  while(pkt_type[type-1] != num_user)
    {
  if(num_track >1)
    index_track = -1;
  index_track++;
  num_track = pkt_type[index_track];
  pkt_type[index_track] = 0;
  pkt_type[0] = num_track - 1;
  pkt_type[index_track + 1]++;
  length++;
  for(j=0; j<type; j++)
    comb[length][j] = pkt_type[j];
    }
  return(length);
}

/* Subroutine realization employs subroutine build_tree */
/* to build a binary tree of realizations. */

struct node *realization(ptr)
struct node *ptr;
{
  int interval;
  int tree_level;
  struct node *build_tree();
  void build_node();

  interval = 1;
  tree_level = 1;
  while(interval<num_interval)
    {
  ptr = build_tree(ptr,interval,tree_level);
  interval++;
    }
  return(ptr);
}
```

```
/* Subroutine build_tree searches for the valid paths of
realizations */

struct node *build_tree(ptr,interval,tree_level)
struct node *ptr ;
int interval;
int tree_level;
  {
    if(tree_level<interval)
      {
    if((ptr->left)->num_intfr != INVALID)
      {
        tree_level++;
                  p  t  r  -  >  l  e  f  t       =
build_tree(ptr->left,interval,tree_level);
        tree_level--;
      }
    if((ptr->right)->num_intfr != INVALID)
      {
        tree_level++;
                  p  t  r  -  >  r  i  g  h  t    =
build_tree(ptr->right,interval,tree_level);
        tree_level--;
      }
      }
    else
      build_node(ptr,interval);
    return(ptr);
  }

/* Subroutine build_node is to fill nodes of a binary */
/* tree with the number of interferers. A left child */
/* node has the number of interferers equal to the */
/* parent's plus one if the parent's is not greater */
/* than n, the maximum number of allowed interferers. */
/* If the parent node is greater than n, the left */
/* child node is set to NULL, i.e. invalid. A right */
/* child node will be equal to the parent's number of */
/* interferers minus one if the parent's is greater */
/* than the minimum number of interferers, which is */
/* equal to number of Early-Late packets. */
/* If it is not, the right child is */
/* set to NULL. The final interval should have the */
/* number of interferes equal to the sum of number */
/* of Early_Late and Late-Late packets. */
/* If it is not, that realization is invalid. */


void build_node(ptr,interval,
struct node *ptr;
int interval;
```

```
   {
     if(ptr->left == NULL)
        {
    if((ptr->left = (struct node *) malloc (NODE_SIZE))
         == NULL)
        {
        printf("ERROR1!!!");
        exit(1);
        }
    (ptr->left)->left = NULL;
    (ptr->left)->right = NULL;
        }
     if(ptr->right == NULL)
        {
    if((ptr->right = (struct node *)malloc(NODE_SIZE))==0)
       {
      printf("ERROR2!!!");
      exit(1);
       }
    (ptr->right)->left = NULL;
    (ptr->right)->right = NULL;
        }
  /* install left child node */
     if(ptr->num_intfr<max_intfr)
        {
    (ptr->left)->num_intfr = ptr->num_intfr + 1;

   /* invalid final interferers */
    if((interval==num_interval-1)&&((ptr->left)->num_intfr
!=final_intfr))
       (ptr->left)->num_intfr = INVALID;
        }
     else
        (ptr->left)->num_intfr = INVALID;

  /* install right child node */
     if(ptr->num_intfr>min_intfr)
        {
       (ptr->right)->num_intfr = ptr->num_intfr - 1;
   /* invalid final interferers */

if((interval==num_interval-1)&&((ptr->right)->num_intfr!
=final_intfr))
       (ptr->right)->num_intfr = INVALID;
        }
     else
        (ptr->right)->num_intfr = INVALID;
  }

/* Subroutine get_cj is to count the number of */
```

```
/* realizations which has the maximum number of */
/* interferes not greater than j. */

void get_cj(ptr)
struct node *ptr;
  {
    int tree_level;
    int max;
    void count_cj();

    tree_level = 1;
    max = ptr->num_intfr;
    count_cj(tree_level,max,ptr);
  }

void count_cj(tree_level,max,ptr)
int tree_level;
int max;
struct node *ptr;
  {
    int old_max;
    if(tree_level<num_interval)
      {
    if((ptr->left)->num_intfr != INVALID)
      {
        tree_level++;
        old_max = max;
        if((ptr->left)->num_intfr>max)
          max = (ptr->left)->num_intfr;
        count_cj(tree_level,max,ptr->left);
        tree_level--;
        max = old_max;
      }
    if((ptr->right)->num_intfr != INVALID)
      {
        tree_level++;
        count_cj(tree_level,max,ptr->right);
        tree_level--;
      }
    }
    else
      cj[max_intfr-1][max-1]++;
    return;
  }

void print_out()
  {
    int total_rlzt;
    double fn;
    int i,j;
for(i=0;i<cj_lgth;i++)
```

```
    {
       fprintf(file,"%d\n",i);
       total_rlzt = 0;
      for(j=0;j<cj_lgth;j++)
        {
          fprintf(file,"%d    " ,cj[i][j]);
          total_rlzt +=  cj[i][j];
        }
      fn = 1.0/(double)total_rlzt;
      fprintf(file,"%d    %8e\n",total_rlzt,fn);
    }
  }
```

B. PRO-MATLAB M-FILE FOR THROUGHPUT OF A RING MODEL

The following M-file is written to run on Pro-Matlab to find the probability of capture and throughput of a ring model with $R_1 = 0$ and $R_2 = 1$.

```
% set w from 1 to infinitive
w = 1:0.1:52.2;
resolution = 0.1;
range = 1/2;
% find the pdf of power of a packet
fw = range.*(w.^(-3/2));
% find j-fold convolution of interfering packets
CONV(1,:) = fw;
CONVFW = fft(fw);
for i= 2:10,
CONV(i,:) = resolution *ifft(fft(CONV(i-1,:)) .* CONVFW);
end
% get f(n) and Cj from data file
data
for i = 1:10,
for j = 1:513,
  tmp(j) = 0;
end
for k = 1:i,
  tmp = tmp + (cj(i,k) .* CONV(k,:));
end
fy(i,:) = fn(i).* tmp;
end
% find fz in fz(z/n) = fz*z^(-3/2)
for i = 1:10,
sum = range * (w(1)^(-0.5)) * fy(i,1);
for j = 2:2:512,
  sum = sum + 4*range*(w(j)^(-0.5))*fy(i,j);
```

```
end
for j = 3:2:511,
  sum = sum + 2*range*(w(j)^(-0.5))*fy(i,j);
end
sum = sum + range*(w(513)^(-0.5))*fy(i,513);
fz(i) = (resolution/3)*sum;
end
% find the probability of a tagged packet
pr = 2 * (6^(-1/2)) .* fz;

% Throughput
g = 0:0.1:5;
for i = 1:51,
s(i) = g(i)/((g(i)+1)^3);
for n = 1:10,
  s(i) = s(i) + (n+1)*(n+2)/2*pr(n) *(g(i)^(n+1))
           * ((g(i)+1)^(-(n+3)));
end
end
```

C. PRO-MATLAB M-FILE TO FIND EQUAL RING THROUGHPUT

This M-file is written for Pro-Matlab to find the equal ring throughputs in chapter 4.

```
% set w from 1 to infinitive
w = 1:0.05:26.6;
resolution = 0.05;
range(4) = 1/2;
% pdf of power of an interfering packet
fw = 1/2.*(w.^(-3/2));
% calculate the j-fold convolution of interfering
% packets
CONV(1,:) = fw;
CONVFW = fft(fw4);
for i= 2:10,
CONV4(i,:)   =   resolution.*   ifft(fft(CONV4(i-1,:))   .*
CONVFW);
end
% get f(n) and Cj from data file
data
% find fy
for i = 1:10,
    for j = 1:513,
       tmp(j) = 0;
    end
for k = 1:i,
    tmp = tmp + (cj(i,k) .* CONV4(k,:));
```

```
end
fy(i,:) = fn(i).* tmp;
end
% polynomial fit for fy(n)
for i = 1:10,
fit(i,:) = polyfit(w,fy(i,:),9);
end
% set range of ring 1 from 0 to 0.52
r1(1) = 0;
r2(1) = 0.52;
range(1) = 1/(2*(0.52^2)) ;
% set range of ring 2 from 0.52 to 0.8
r1(2) = 0.52;
r2(2) = 0.8;
range(2) = 1/(2*(0.8^2 -0.52^2));
% set range of ring 3 from 0.8 to 1
r1(3) = 0.8;
r2(3) = 1;
range(3) = 1/(2*(1 - 0.8^2));
% find fz in fz(z/n) = fz*z^(-3/2)
for ring = 1:3,
for i = 1:10,
for z = 6:.1:24.4,
  if ring == 3,
  start = 1;
  else
  start = r2(ring)^(-4)/z;
  end
  if ring ==1,
  ends = 26.6;
  else
  ends = r1(ring)^(-4)/z;
  end
  temp = 0;
  if start < 1,
   start = 1;
  end
  if ends > 26.6,
  ends = 26.6;
  end
  if ends > 1,
  step = (ends - start)/8;
  y = start:step:ends;
  clear f;
  f = polyval(fit(i,:),y);
  sum = range(ring) * (y(1)^(-1/2)) * f(1);
  for j = 2:2:8,
  sum = sum + 4*range(ring)*(y(j)^(-0.5))*f(j);
  end
  for j = 3:2:7,
  sum = sum + 2*range(ring)*(y(j)^(-0.5))*f(j);
```

```
  end
  sum = sum + range(ring)*(y(9)^(-0.5))*f(9);
  temp = (step/3)*sum;
  end
  if z == 6,
     pr(ring,i) = temp * z^(-3/2);
  elseif z < 13.7,
     pr(ring,i) = pr(ring,i)+ 2*temp*z^(-3/2);
  elseif z == 13.7,
     pr(ring,i) = pr(ring,i)+temp*z^(-3/2) ;
  end
end
pr(ring,i) = pr(ring,i)*.1/2;
end
% calculate p(3,i) for z = 13.7 to infinitive.
  if ring == 1,
  for i = 1:10,
  y = 1:.1:26.6;
  clear f;
  f = polyval(fit(i,:),y);
  sum = range(ring) * (y(1)^(-1/2)) * f(1);
  for j = 2:2:256,
  sum = sum + 4*range(ring)*(y(j)^(-0.5))*f(j);
  end
  for j = 3:2:255,
  sum = sum + 2*range(ring)*(y(j)^(-0.5))*f(j);
  end
  sum = sum + range(ring)*(y(257)^(-0.5))*f(257);
  temp = (.1/3)*sum;
  temp = 2*(13.7^(-1/2))*temp;
  pr(ring,i) = pr(ring,i)+temp;
  end
  end
% Throughput
g = 0:0.05:4;
for i = 1:81,
s(ring,i) = 1/(2*range(ring))*g(i)/((g(i)+1)^3);
for n = 1:10,
  s(ring,i) = s(ring,i) + (n+1)*(n+2)/2*pr(ring,n)
  /(2*range(ring))*g(i)*(g(i)^(n))*((g(i)+1)^(-(n+3)));
end
end
end
```

APPENDIX B. COMPUTATION OF CONDITIONAL CAPTURE PROBABILITY IN RING R_i

Let us consider a tagged packet in ring R_i bounded by two concentric circle of radii r_i and r_{i+1}, respectively. The conditional probability density function for packet generation in ring R_i is given by

$$f_R(r|R_i) = \begin{cases} \dfrac{f_R(r)}{F_R(r_{i+1}) - F_R(r_i)} & , \; r_i \le r \le r_{i+1} \\ 0 & , \; otherwise \end{cases} \tag{B1}$$

The probability a packet is generated within a distance $r \le r_i$ is

$$F_R(r_i) = \frac{2\pi}{G}\int_0^{r_i} x \frac{G}{\pi(R_2^2 - R_1^2)} dx \tag{B2}$$

Solving (B2), we have

$$F_R(r_i) = \frac{r_i^2}{R_2^2 - R_1^2} \tag{B3}$$

Similarly,

$$F_R(r_{i+1}) = \frac{r_{i+1}^2}{R_2^2 - R_1^2} \tag{B4}$$

Substituting (B3) and (B4) into (B1), we obtain the pdf of packet generation in ring R_i as

$$f_R(r|R_i) = \begin{cases} \dfrac{2r}{r_{i+1}^2 - r_i^2} & , \quad r_i \le r \le r_{i+1} \\ 0 & , \quad otherwise \end{cases} \tag{B5}$$

The corresponding conditional pdf for packet power X in ring R_i can be obtained from (16) as

$$f_X(x|R_i) = \begin{cases} \dfrac{x^{-(\frac{3}{2})}}{2(r_{i+1}^2 - r_i^2)} & , \quad r_i \le r \le r_{i+1} \\ 0 & , \quad otherwise \end{cases} \tag{B6}$$

The pdf of the random power of n interfering packets anywhere in the network is obtained from (19). Therefore, the conditional pdf of signal to interference ratio in ring R_i is

$$f_Z(z|n,R_i) = \int_0^\infty y f_X(yz|R_i)\, f_Y(y|n)\, dy \tag{B7}$$

Then the conditional capture probability in ring R_i is

$$Pr\{capture|n,R_i\} = \int_{\gamma_o}^\infty f_Z(z|n,R_i)\, dz$$

APPENDIX C. DERIVATION OF THROUGHPUT OF A RING IN A SYSTEM OF MULTIPLE RINGS.

The derivation of the ring throughput for a multiple ring system is similar to the derivation in section II.B.

The channel throughput S_i, in packets per mean packet time becomes [5]

$$S_i = \int_0^{\infty}\int_0^{\infty}\frac{1}{2}\phi_i(x,u)\,a(x)\,a(u)\,dxdu \tag{C1}$$

where a(x) and a(u) are exponential density functions with mean $\overline{x}$ of the tagged packet length and the preceding packet length. The quantity $\phi_i(x,u)/2$ is the conditional channel departure rate in ring R_i [5].

$$\frac{1}{2}\phi_i(x,u) = \frac{1}{2}g_i(x+u)\,Pr(capture) \tag{C2}$$

where g_i is a parameter in packets per second of a Poisson process of the channel rate in ring R_i.

Substituting (26) and (C2) into (C1), we have

$$S_i = \int_0^{\infty}\int_0^{\infty}\frac{1}{2}g_i(x+u)\sum_{n=0}^{\infty} e^{-g(x+u)}\frac{[g(x+u)]^n}{n!} \cdot Pr\{capture|n\}\,a(x)\,a(u)\,dxdu \tag{C3}$$

Utilizing the binomial identity

$$(x+u)^n = \sum_{k=0}^{n}\binom{n}{k}x^k\,u^{n-k} \tag{C4}$$

and expanding (C3), we obtain

$$S_i = \frac{1}{2}g_i\int_0^{\infty}\int_0^{\infty}[xe^{-gx}\,e^{-gu} + ue^{-gu}\,e^{-gx}]\sum_{n=0}^{\infty}\frac{g^n}{n!}Pr\{capture|n\} \cdot \sum_{k=0}^{n}\binom{n}{k}x^k\,u^{n-k}\,a(x)\,a(u)\,dxdu \tag{C5}$$

Further manipulating terms in the above equation, we get

$$S_i = \frac{1}{2}g_i\sum_{n=0}^{\infty}\frac{g^n}{n!}Pr\{capture|n\} \cdot \left[\sum_{k=0}^{n}\binom{n}{k}\int_0^{\infty}x^{k+1}e^{-gx}a(x)\,dx.\int_0^{\infty}u^{n-k}e^{-gu}a(u)\,du + \sum_{k=0}^{n}\binom{n}{k}\int_0^{\infty}u^{n-k+1}e^{-gu}a(u)\,du.\int_0^{\infty}x^{k}e^{-gx}a(x)\,dx\right] \tag{C6}$$

We have assumed that the packet length is exponentially distributed and the average length is $\bar{x}$. Hence

$$a(x) = \frac{1}{\bar{x}}e^{-x/\bar{x}} \tag{C7}$$

$$a(u) = \frac{1}{\bar{x}}e^{-u/\bar{x}} \tag{C8}$$

Define the average traffic rate G of the system as

$$G = g\bar{x} \tag{C9}$$

We apply (C7), (C8), (C9) and the following equation [9].

$$\int_0^{\infty}x^n\,e^{-\mu x}\,dx = n!\,\mu^{-(n+1)} \tag{C10}$$

to solve (C6)

The first integral term can be evaluated as

$$\int_0^\infty x^{(k+1)} e^{-gx} a(x)\,dx = \frac{1}{\bar{X}}\int_0^\infty x^{(k+1)} e^{-(g+\frac{1}{\bar{X}})x} dx$$
$$= \frac{(k+1)!}{\bar{X}} \left(g+\frac{1}{\bar{X}}\right)^{-(k+2)} \tag{C11}$$
$$= \frac{(k+1)!}{\bar{X}^{-(k+1)}} (G+1)^{-(k+2)}$$

Similarly, the second integral is found to be

$$\int_0^\infty u^{(n-k)} e^{-gu} a(u)\,du = \frac{1}{\bar{X}}\int_0^\infty u^{(n-k)} e^{-(g+\frac{1}{\bar{X}})u} du$$
$$= \frac{(n-k)!}{\bar{X}} \left(g+\frac{1}{\bar{X}}\right)^{-(n-k+1)} \tag{C12}$$
$$= \frac{(n-k)!}{\bar{X}^{-(n-k)}} (G+1)^{-(n-k+1)}$$

The third integral term is

$$\int_0^\infty u^{(n-k+1)} e^{-gu}\, a(u)\,du = \frac{(n-k+1)!}{\bar{X}^{-(n-k+1)}} (G+1)^{-(n-k+2)} \tag{C13}$$

The final integral term is reduced to

$$\int_0^\infty x^n e^{-gx} a(x)\,dx = \frac{k!}{\bar{X}^{-k}} (G+1)^{-(k+1)} \tag{C14}$$

Combining all integral terms in (C11), (C12), (C13), and (C14) into (C6), we obtain

$$S_i = \frac{1}{2} g_i \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\} \frac{(G+1)^{-(n+3)}}{\bar{X}^{-(n+1)}} \sum_{k=0}^{n} n!\,(n+2) \tag{C15}$$

Applying (C9), the equality

$$\sum_{k=0}^{n} n!\,(n+2) = n!\,(n+1)\,(n+2) \tag{C16}$$

nd the fact that $G_i = g_i\bar{x}$, and further simplifying (C15), we obtain

$$S_i = \sum_{n=0}^{\infty} \frac{(n+1)(n+2)}{2} Pr\{capture|n\} G_i G^n (G+1)^{-(n+3)} \qquad \text{(C17)}$$

APPENDIX D. DERIVATION OF PROBABILITY OF CAPTURE.

The probability of capture in a ring Pr{capture}, per mean packet time can be found by

$$Pr\{capture\} = \int_0^\infty \int_0^\infty \sum_{n=0}^\infty Pr\{capture|n\}\, Pr\{n\} \cdot a(x)\, a(u)\, dx du \tag{D1}$$

where a(x) and a(u) are exponential density functions with mean $\bar{x}$ of the tagged packet length and the preceding packet length.

Substituting (26) into (D1), we have

$$Pr\{capture\} = \int_0^\infty \int_0^\infty \sum_{n=0}^\infty e^{-g(x+u)} \frac{[g(x+u)]^n}{n!} \cdot Pr\{capture|n\}\, a(x)\, a(u)\, dx du \tag{D2}$$

Utilizing the binomial identity

$$(x+u)^n = \sum_{k=0}^{n} \binom{n}{k} x^k\, u^{n-k} \tag{D3}$$

and expanding (D2), we obtain

$$Pr\{capture\} = \int_0^\infty \int_0^\infty \sum_{n=0}^\infty \frac{g^n}{n!} Pr\{capture|n\}\, e^{-g(x+u)} \cdot \sum_{k=0}^{n} \binom{n}{k} x^k\, u^{n-k}\, a(x)\, a(u)\, dx du \tag{D4}$$

Further manipulating terms in the above equation, we get

$$Pr\{capture\} = \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\}$$
$$\cdot \sum_{k=0}^{n} \binom{n}{k} \int_0^{\infty} x^k e^{-gx} a(x)\, dx \cdot \int_0^{\infty} u^{n-k} e^{-gu} a(u)\, du \tag{D5}$$

We have assumed that the packet length is exponentially distributed and the average length is $\overline{x}$. Hence

$$a(x) = \frac{1}{\overline{x}} e^{-x/\overline{x}} \tag{D6}$$

$$a(u) = \frac{1}{\overline{x}} e^{-u/\overline{x}} \tag{D7}$$

Define the average traffic rate G of the system as

$$G = g\overline{x} \tag{D8}$$

We apply (D6), (D7), (D8) and the following [9]

$$\int_0^{\infty} x^n\, e^{-\mu x}\, dx = n!\,\mu^{-(n+1)} \tag{D9}$$

to solve (D5)

The first integral term can be evaluated as

$$\begin{aligned} \int_0^{\infty} x^k e^{-gx} a(x)\, dx &= \frac{1}{\overline{x}} \int_0^{\infty} x^k e^{-(g+\frac{1}{\overline{x}})x} dx \\ &= k\frac{!}{\overline{x}} \left(g+\frac{1}{\overline{x}}\right)^{-(k+1)} \\ &= k\frac{!}{\overline{x}^{-k}} (G+1)^{-(k+1)} \end{aligned} \tag{D10}$$

Similarly, the second integral is found to be

$$\int_0^{\infty} u^{(n-k)} e^{-gu} a(u)\, du = \frac{(n-k)!}{\overline{x}^{-(n-k)}} (G+1)^{-(n-k+1)} \tag{D11}$$

Combining all integral terms in (D10), (D11), into (D5), we get

$$Pr\{capture\} = \sum_{n=0}^{\infty} \frac{g^n}{n!} Pr\{capture|n\} \frac{(G+1)^{-(n+2)}}{\overline{X}^{-n}} \sum_{k=0}^{n} n! \quad (D12)$$

Applying (D8) and the equality

$$\sum_{k=0}^{n} n! = n!(n+1) \quad (D13)$$

and further simplifying (D12), we obtain

$$Pr\{capture\} = \sum_{n=0}^{\infty} Pr\{capture|n\}(n+1)G^n(G+1)^{-(n+2)} \quad (D14)$$

LIST OF REFERENCES

1. Robertson, R. C., and Ha, T. T., "A Model for Local/Mobile Radio Communications with Correct Packet Capture," IEEE Trans Comm., accepted for publication.

2. Goodman, D. J., and Saleh A. A. M., "The Near/Far Effect in Local Aloha Radio Communications," IEEE Trans. Veh. Tech., vol. VT-36, no. 1, pp.19-27, Feb. 1987.

3. McCartin, J. T., and Ha, T. T., "Ring Model for Local Radio Communications," Globecom '89 Dallas, TX, pp. 20.2.1-20.2.5.

4. Abramson, N., "The Throughput of Packet Broadcasting Channels," IEEE Trans. Commun., vol COM-25, no. 1, pp. 117-128, Jan. 1977.

5. Ha, T. T., "Aloha VSAT Networks with Data Link Control," ICC '88, pp. 25.3.1-25.3.4.

6. McCartin, J. T., and Ha, T. T., "Variable Packet Aloha Networks," NTC '90, pp. 360-365.

7. Nijenhuis, A., and Wilf, H. S, Combinatorial Algorithms, Academic Press, 1975.

8. Garcia, A. L., Probability and Random Processes for Electrical Engineering, p. 229, Addison-Wesley Publishing Co., 1989.

9. Gradshteyn, I. S., and Ryzhik, I. M., Table of Integrals, Series, and Products, Academic Press, 1980.

10. Schwartz, M., Telecommunication Networks Protocols, Modeling and Analysis, pp. 423-433, Addison-Wesley Publishing Co., 1988.